FACULDADE DE ENGENHARIA DA UNIVERSIDADE DO PORTO

Real Estate Market Data Scraping and Analysis for Financial Investments

João Manuel Azevedo Santos

Mestrado Integrado em Engenharia Informática e Computação

Supervisor: Pedro Gonçalo Ferreira Alves Nogueira

September 10, 2018

Real Estate Market Data Scraping and Analysis for Financial Investments

João Manuel Azevedo Santos

Mestrado Integrado em Engenharia Informática e Computação

September 10, 2018

Abstract

All around the world, the real estate market attracts hundreds of millions of euros every year. Despite some major disruptions in the context of the financial crisis, the real estate market remains as one of the major and most stable markets suitable for investments at various scales. Big investment groups and institutions have the knowledge and resources to gather and analyze all sorts of information regarding this market. On the other hand, it is hard for individual investors to make informed decisions about assets in which to invest in because of the lack of knowledge and resources.

The information that exists in current real estate online platforms provides low value for investors looking to make an informed decision. Often, relevant information is not disclosed and there are no indicators of property value and its fluctuations. This lowers the possibility of success of the individual investor. The investor is forced to perform intensive hands-on research in order to analyze all the important details or ends up making decisions based in tendencies and/or marketing materials. All in all, the retrieval of this information seems to be challenging because it exists in several different locations and is presented and structured in distinct ways.

Considering the problem at hand, this project proposes the development of a web scraper for the selected sources of relevant data. By cleaning, storing and modeling this data in a flexible data structure it will enable the development of an online aggregation platform for real estate market data. On top of the normal search and listing criteria that current platforms allow for, investors will also have access to price analysis by time, area and location. The platform shall also present the results of a classification task that was performed by extracting relevant features from the data gathered. The model built with these aims to predict the fluctuations of the asset's prices.

This solution will enable a deeper understanding of the real estate market landscape and provide a unique, centralized and insightful source of information for potential investors.

Resumo

O mercado imobiliário atrai anualmente centenas de milhões de euros por todo o mundo e apesar de algumas quebras em contexto de crises financeiras mantém-se até aos dias de hoje como um dos maiores e mais estáveis mercados de investimento, adequado para investidores de todas as dimensões. Os grandes grupos e instituições de investimento possuem os conhecimentos, as técnicas e os recursos que lhes permitem recolher e analizar todo o tipo de informação relevante neste mercado. Por outro lado, é difícil para o investidor individual tomar decisões bem informadas devido quer ao desconhecimento quer à inexistência de informação relevante.

A informação existente nas plataformas online do ramo imobiliário apresenta pouco valor para os investidores. Informação importante e relevante é frequentemente inexistente e não existem indicadores de valor das propriedades e da sua variação. Isto torna mais difícil o sucesso do investidor, pois este é forçado a tomar as suas decisões com base em tendências e materiais de marketing e/ou conduzir uma pesquisa, compilação e análise extensas e demoradas, de forma a conseguir analisar todos os detalhes importantes. Analisando todos estes fatores, a recolha desta informação apresenta diversos desafios: encontra-se dispersa em diversas plataformas com diferentes formas de apresentar e estruturar o seu conteúdo e podem ainda existir diversos problemas que afetem a qualidade dos dados.

Neste projeto, propõe-se então o desenvolvimento de web scrapers para as diversas fontes de dados selecionadas. Após tratamento, limpeza, transformação e armazenamento desta informação será possível construir uma plataforma que agrega todos os dados recolhidas sobre possíveis investimentos neste mercado. Para além das funcionalidades de listagem e busca que as plataformas atuais oferecem, os investidores terão ainda acesso a análises da variação do preço por tempo, área e localização. A plataforma disponibilizará ainda os resultados de uma tarefa de classificação que foi realizada extraindo atributos relevantes da informação recolhida para treinar um modelo de forma a tentar prever as flutuações de preço dos imóveis.

A solução apresentada permitirá um aprofundamento do conhecimento sobre o mercado imobiliário e oferece aos investidores uma fonte de informação única, centralizada e detalhada.

Acknowledgements

First of all, my deepest feelings of gratitude go towards my parents and sister. Without their love, guidance, trust and support to lift me up in the bad moments I would not have been able to achieve such good things in my life.

I would also like to stress my great respect and appreciation for my supervisor, Professor Pedro Gonçalo Ferreira Alves Nogueira, for his invaluable support and availability to guide me and help me overcome the difficulties throughout the development of this dissertation.

Thirdly, to all my friends and companions in this journey at FEUP. For all the help, guidance, friendship and companionship and all the great moments spent together, my most sincere thanks for everything.

Lastly, a note of appreciation to all the authors referenced throughout this work for their commitment and contributions. Standing on their "shoulders of giants" I hope to have given some contribution myself.

João Santos

"One man's "magic"
is another man's engineering."

Robert A. Heinlein

Contents

List of Figures

LIST OF FIGURES

List of Tables

Abbreviations

ANN	Artificial Neural Network
API	Application Programming Interface
DBSCAN	Density-Based Spatial Clustering of Applications With Noise
DM	Data Mining
DOM	Document Object Model
HTTP	Hypertext Transfer Protocol
IP	Internet Protocol
KDD	Knowledge Discovery from Data
MVC	Model View Controller
NPM	Node Package Manager
OPAC	Online Public Access Catalogues
UI	User Interface
WWW	*World Wide Web*

Chapter 1

Introduction

In 2017, more than twenty thousand million euros were invested in the real estate market. Despite some occasional disruptions, this market has maintained itself throughout the years as one of the most solid and well established investment markets.

Current web platforms for real estate, advertise the selling and renting of real estate assets and provide the means for searching and listing different categories but fail to provide investors with essential historical data and overviews of price analysis and variation.

By providing all these features in a single centralized aggregating platform, investors and even regular buyers will have a valuable tool to help them better decide where and how to invest their money.

1.1 Context and motivation

Throughout the years the real estate market has presented many investment opportunities for investors at various levels all around the world. The European real estate market alone attracts hundreds of millions of euros every year. As shown in Figure 1.1 total investment volumes since 2013 have maintained relatively high and consistent levels of investment. This is despite some years presenting much better results than others, which may be linked to economic prosperity or stagnation.

According to the most recent reports, despite some challenges such as the lack of income growth, it appears that this market remains as one of the most active, and a steady growth is expected particularly in Europe and Japan. This supports the statement that real estate is still, very much in demand. Investors and buyers want to make informed decisions regarding their financial investments. The main reasons that they present for investing in real estate can be divided in two different categories: financial and social. On the first, they look for ways to maximize their earnings by investing in properties with high value in terms of current and future profitability. On the latter, investors are mainly interested in stable and peaceful political and social environments.

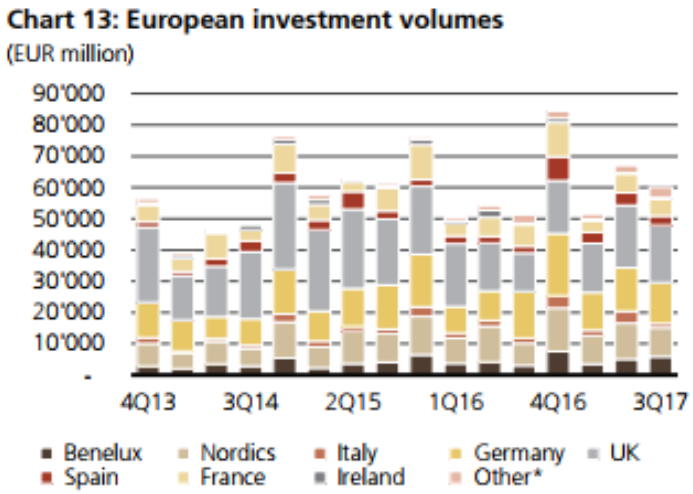

Figure 1.1: European investment volumes

In a survey conducted in 2017 with European investors, the researcher's aim was to find the main motivations for investing in real estate. Their findings are presented in Figure 1.2

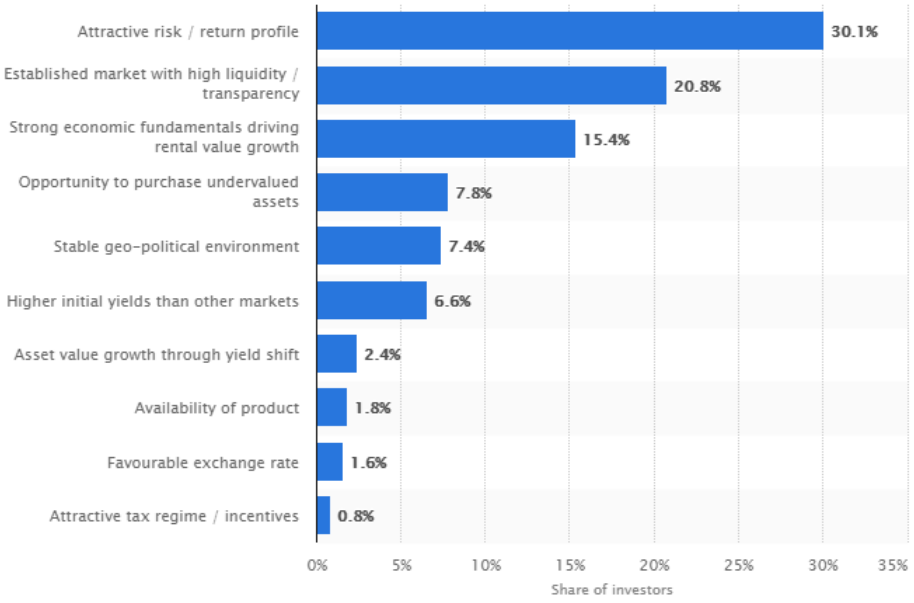

Figure 1.2: Main motivations for investing in real estate in Europe, 2017

We can see that for investors, economic and financial aspects are the most valuable. They look for attractive risk / return profiles, in well established markets with high liquidity and strong economic fundamentals that drive growth of property and/or rent values.

1.2 Problem description

Current web platforms provide little information with low value on its own such as the current price, location, typology and images. This information alone is not sufficient for investors to analyze and predict the profitability of their real estate assets as they don't have access to the variation of price of the asset itself or the neighbourhoods'. This data is spread throughout many different web platforms, each with its own way of structuring and presenting the information. There might also be problems affecting the quality of the data such as missing values, redundancies, inconsistencies and noise. This data will be collected by using automated mechanisms. Even if there are missing values around price variation a dataset can start to be built through these mechanisms.

After compiling this information we should be able to provide the investors with a centralized and insightful way to search for and analyze their potential real estate investments.

1.3 Goals

In this project we set to develop several automatic mechanisms for collecting, updating and unifying real estate data from the selected data sources. These can be images, keywords, dates, area, typology, location, contacts, price or others. We are particularly interested in information about the location, price, area, photos and time-stamps as these will serve as a base model with lots of potential for extracting the knowledge we want. For this we intend to perform several price analyses by area, location, time and other existing categories.

Another important goal of this project is the creation of a labeled dataset regarding fluctuation of asset prices. Using the automated update mechanisms mentioned before we can track if the price of a certain asset went up or down and use that information to perform an automated labeling task of our dataset. The labeled dataset will be used to perform a classification task in the scope of the current project. However, it can also have other applications and uses, and other data scientists may use it to perform their own descriptive and predictive tasks.

Finally, we will create interactive dashboards with our price analysis and prediction results to present this information to the investors through the means of a web platform. This platform will provide listing and searching capabilities by different criteria and characteristics as well as the visualization of the interactive dashboards as previously mentioned.

1.4 Document structure

Besides Chapter 1, this document contains 5 other chapters. On chapter 2, the state of the art for our problem is described and some related works are presented. After a brief introduction , the main areas of interest are presented starting with web scraping as a sub field of information retrieval. Then we focus on the more recent data analysis, data mining, visualization tools and techniques used in the real estate market context. Finally, the main conclusions of this study are presented in the last section. Chapter 3, describes the implemented solution for the scraping

application. After stating its purpose, the methodology put in practice is presented followed by an overview of the technologies used. The system's architecture is then presented and explained and this chapter ends with a presentation and discussion of the main challenges faced, how they were overcome and the limitations of the system built. Chapter 4 presents the prediction task that was carried out with the real estate data gathered. After describing the task and technologies selected to perform it, the dataset is thoroughly described followed by the methodology applied for labeling it. Then, the process of preprocessing the data is described and all of its phases are presented individually. After this we present the different experiments performed to train the models with different algorithms and parameters. The main results and conclusions are also presented here. The chapter ends with a brief reflection on the task's results and how they could be improved. Chapter 5 describes the web platform built, to search for real estate assets and visualize the results of the analysis performed on the data retrieved. After a brief introduction the technologies used are presented, followed by the system's architecture, which is described in its components, front-end and back-end. Finally the platform views are presented and the chapter ends with a brief conclusion and discussion about the limitations of the platform. Lastly, chapter 6 presents the global conclusions of the work developed and the future work that could benefit the solution to provide even more value.

Chapter 2

Literature Review

2.1 Introduction

In this chapter we will explore the state of the art on information retrieval from the web, focusing on data scraping and the more adequate tools for performing this sort of tasks. We will also review the most commonly used methods for real estate data analysis using data mining methods and predictive algorithms. The visualization of this data will also be studied and explored. This has been a fast-growing field in the last years as a result of a large availability of information, the capability to process a lot of data and the necessity to present the knowledge extracted from raw data to non-specialized subjects. All these will be of major importance for the development of the proposed solution, for better understanding of the scope of the problem and for recognizing the best tools and technologies to achieve our goals.

2.2 Information retrieval

Information retrieval from the web has been a topic of concern for researchers for a very long time. In the era of big online platforms and communities like YouTube, Facebook and even electronic commerce stores like Amazon, the usage of information retrieval systems is now in higher demand than it has ever been. As we search for our favourite song, browse for an acquaintance online or try to find the best recipe for preparing dinner we are making use of these kind of systems. As a result, there has been a great amount of research on these topics.

In a recent research research study, the authors present the broad outline of an information retrieval system and identify different ways to categorize different information retrieval systems. This general overview of these systems is presented in Figure 2.1.

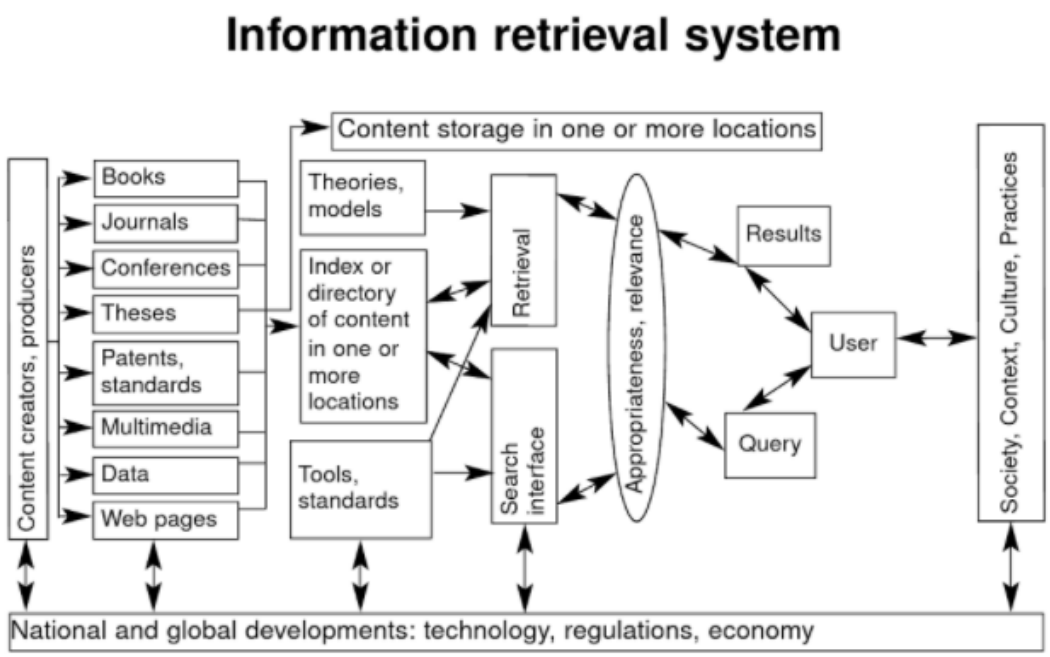

Figure 2.1: Information Retrieval System Overview. [Cho10]

They propose a different approach to the grouping of the information retrieval systems, based on their content, purpose and functions. In this approach four different types of information retrieval systems can be identified: Online Public Access Catalogues (OPAC), online databases, digital libraries and web-based information services and web search engines [Cho10].

An important characteristic of today's database systems is that they are web-based. Anyone with access to a common browser can access and query these databases through simple inputs and without the need of expertise on these topics.

2.2.1 Web scraping

The search and extraction of information from the World Wide Web (WWW) is usually performed by web crawlers. A web crawler is a computer program or an automated script that browses the Web in a structured and automated manner. Web scrapers are a type of Web crawler that aims to find specific kinds of information and aggregate it for the production of new content. [KT00].

Web scraping (also called Web harvesting or Web data extraction) is a software technique aimed at extracting information from websites [Sch12]. Web scrapers try to automate the exploration process of the WWW, by either implementing the hypertext transfer protocol (HTTP) or embedding suitable Web browsers. The data is usually gathered through screen outputs or directly from the HTML of the page. Web scraping focuses on the transformation of unstructured data on the Web, typically in HTML format, into structured data that can be stored and analyzed in a central local database or spreadsheet[VU12].

There are many methods to scrape information from the Web[HLA15]. Human copy-paste is the most straightforward method, but it proves unfeasible due to the high cost of resources especially in big projects. Another method used is text grepping, in which the programs try to match regular expressions with the patterns they expect to find. Other techniques are HTTP programming, DOM parsing, and HTML parsers[HLA15]. In the advent of Big Data, the potential applications for scraped data and the benefits associated with its analysis have increased exponentially. Web scraping is currently used for online price comparison, weather data monitoring, website change detection, Web research, Web mash-up, and Web data integration. Several (commercial or open-source) tools, aimed at personalizing websites by adopting scraping techniques are currently available[VU12].

The following sections will present a brief review of the available tools and technologies which make use of web scraping methods.

2.2.1.1 Beautiful soup

Beautiful Soup is a Python library for fetching data from HTML files. It provides a set of functions to help programmers navigate, search and modify the DOM tree. This is open-source software.

2.2.1.2 Selenium

Selenium is a browser automation software. It is most commonly used for its automated testing capabilities but it can also serve as a scraping tool. It allows to search and navigate through webpages and retrieve any content that they contain. It is also open-source software.

2.2.1.3 Screen-scraper

Screen-Scraper is a powerful tool for extracting data and information from the web. It offers a friendly interface that eases the process of building the data patterns and creating crawling mechanisms with a simple point-and-click interface. It allows the retrieval of data from all kinds of web platforms and to export the results to a variety of formats.[HKD13].

2.3 Real estate data analysis

Real estate data analysis for investment purposes, only started to appear in real estate literature in the late 1960s and early 1970s, even though there were already several analytic tools and techniques used by economists and financial analysts. Up until then terms such as utility functions, profitability and time value of money were nonexistent in the real estate appraisal and investment literature. [RS70]

More recent analyses involve modern decision-theory concepts and complex financial equity evaluation techniques. With the wide adoption of computer-based systems it is now possible to model and explore the impact of a variety of factors on the value of assets.

State-of-the-art real estate analyses approach real estate investment as a capital asset which is able to generate its own stream of profits. Therefore, real estate can be regarded as a case of capital budgeting. This process, consists in determining and evaluating all the future costs and investments needed in the long-term. The cash-flows at various stages of the investment project are estimated to determine whether the potential returns generated meet a sufficient target benchmark. [GK03]

2.3.1 Knowledge discovery

Knowledge discovery from data (KDD), also referred to as data mining (DM) is a prominent and fast-growing field in data science. The advent of the WWW and all of its web platforms, such as YouTube, Facebook, internet banking, real estate searching, medical sites and many others, has led to massive amounts of raw data being stored by private companies and government agencies all over the world. Storing this data is very cheap considering the potential value of the data along with the decreasing prices of storage systems and rise in their capacity. However, the the main problem that modern data scientists are faced with is that the true value is not in the raw data stored but in their ability to extract meaningful relations, patterns, trends and correlations in said data. [KM06]. KDD is by nature interdisciplinary and has many different applications, such as:

- Marketing - Making use of the database marketing systems, the data stored can be used to identify different customer groups and forecast their needs. Another very interesting application is called market-basket analysis, which attempts to find patterns in shopping behaviours such as if a customer bought a specific item then he is also likely to buy some other specified item. This information is of major importance to businesses as they can recommend and expose their customers to the products that they are more likely to buy [FPSS96].

- Investment - Many different companies build their own data mining models for investment processes, however most don't disclose how they are built. In a recent paper from [CCMO14], the scientists developed a system based on neural networks for real estate price appraisal taking into account environmental factors of the property location.

- Fraud Detection - Data mining systems have been used for monitoring credit card fraud, watching over millions of accounts and pinpointing certain financial transactions that may hint at the occurrence of money laundering activity. [SGW+95].

- Manufacturing - Data mining plays a very important role in modern manufacturing engineering. It has several applications in production processes, fault detection, operations optimization, product quality improvement, maintenance and decision support. [HSK06]

- Telecommunications - Big telecommunications companies produce a great amount of information. Data mining techniques may be applied to improve marketing effectiveness, identify network faults and telecommunications fraud. [Wei05]

Some data is already collected and made available in structured datasets, organized in tabular format where each row represents an instance and each column represents an attribute, that are ready to be used in knowledge discovery tasks. Platforms such as *Kaggle*, provide free datasets and promote data mining competitions where data scientists are encouraged to find their own strategies and develop their own solutions.

2.3.1.1 Most common data mining tasks

In the DM field, it is common to categorize the tasks in two different categories:

- Descriptive analytics - Summarizing and aggregating data for pattern extraction. In descriptive tasks the results are obtained by directly applying an algorithm to the data.
- Predictive analytics - Extracting a model from the data to be used in predictions. By using a set of predictive attributes, also known as features, a model is induced which labeling of new unlabeled objects.

[Kan11], identifies in his book the main DM tasks:

- Classification - Discovering of a learning function that can classify an item into one of different classes.
- Regression - Aims to assign a quantitative value to an unlabeled item, given the value of its attributes.
- Clustering - A predictive task that fits items into a finite set of groups where items in the same group are more similar among themselves than data from different groups.
- Summarization - Seeks to summarize and aggregate the data in order to obtain a more compact dataset.
- Dependency Modeling - Finding relations between values of attributes and/or the predictive attributes themselves.
- Change and deviation detection - Detecting the main modifications in the data.

2.3.1.2 The KDD process

The KDD process is defined by [FPSS96] as the process of using DM techniques for extracting knowledge from raw data. This process is described in 5 stages:

- Selection - Consists in the identification and selection of all relevant data sources, focusing on target samples or variables that will be used throughout the KDD process.

- Preprocessing - The data gathered from the selected sources is often noisy, unstructured data and prone to inconsistencies. Therefore, a set of operations needs to be performed to ensure the quality of the data. This includes removing noise or outliers if appropriate, deciding on how to properly handle missing data, and how to make use of temporal information. There are also issues related to data types, schema, and mapping of missing and unknown values.
- Transformation - This step consists in converting the data into suitable formats for DM algorithms.
- Data Mining - In this stage, specialized tools with specialized algorithms are used to find patterns in the data. At the end of the DM process there should be a report of the analysis performed so that the results can be verified and evaluated.
- Interpretation/Evaluation - This consists of the interpretation and analysis of DM process results. This evaluation should be made in conjunction with experts and business analysts from the specific research field, so that if the results are not good enough a new iteration of the process can be initiated.

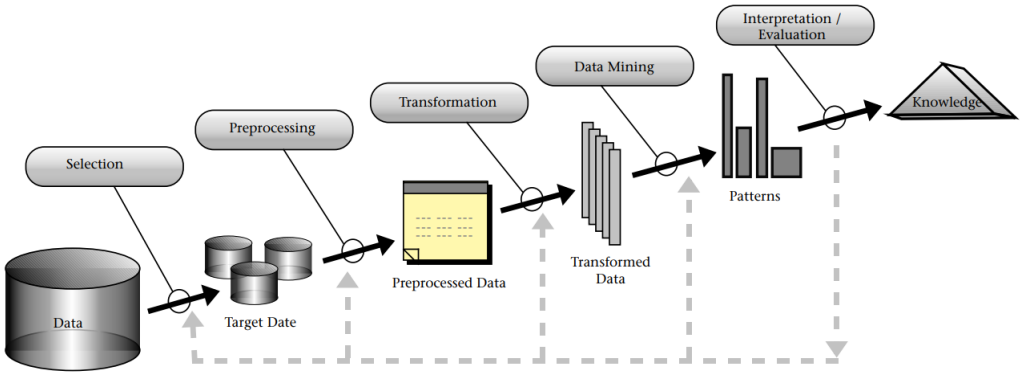

Figure 2.2: The five stages of KDD. [FPSS96]

2.3.1.3 The SEMMA process

The SEMMA process was developed by the SAS institute. It proposes a cycle with 5 stages that should conduct the life cycle of a data mining project [AS08]:

- Sample - Extraction of a data sample that is big enough to contain all the significant information and small enough that it can be analyzed and manipulated quickly.
- Explore - Exploration of the data to find unexpected patterns and/or correlations that are not evident to the naked eye.

- Modify - Modification of the data by selecting, creating and transforming the variables.
- Model - Allowing an algorithm to automatically create a model that consistently predicts a desired outcome.
- Assess - Evaluating the usefulness, reliability and correctness of the findings.

This process offers a well detailed and organized process to develop and maintain DM projects. It helps with finding and solving the business problems as well as implementing new business goals [SA05].

2.3.1.4 The CRISP-DM process

CRISP-DM stands for Cross-Industry Standard Process for Data Mining. It was developed by an industry consortium to address the necessity of standardizing DM practices. It is a very complex process and it requires specialists with proper tools. Hence, the emphasis is in properly managing all the aspects of the project and following a sound methodology [WH00].

- Understanding the business - Focuses on defining the purposes of the project from an organizational and business perspective.
- Understanding the data - Starts with getting acquainted with the data and its intricacies while exploring data quality issues and finding interesting sets.
- Data preparation - Consists of transforming the original dataset in the one that is going to be fed into the modeling tool. It includes a range of operations that can be performed including: cleaning the data, replacing missing values, normalization and aggregating attributes to obtain new features.
- Modeling - In this step apply various modeling techniques and optimize its parameters. Going back and forth between this phase and the data preparation is common.
- Evaluation - At this stage, a moderately high-quality model should have been obtained and we should ensure that the followed process accurately reflects business objectives.
- Deployment - The last step is to structure, model and present the knowledge discovered to the customer.

The sequence of steps of CRISP-DM is not a strict one, as we can understand by analyzing 2.3. It is common for new models and results to drive other business objectives and therefore, restart the whole cycle. This process is very well documented, studied and organized, and it allows for an easier understanding and revising of DM projects.

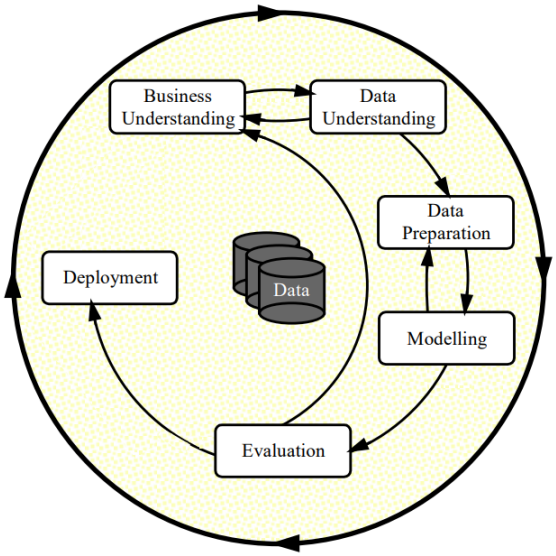

Figure 2.3: Phases of the CRISP-DM process. [WH00]

2.3.1.5 Summary

Although the stages on the different processes seem to have direct correspondence on the others, the CRISP-DM process introduces the business understanding and deployment phases as intrinsic and imperative to the DM proccess. It has been argued that KDD and SEMMA, with their Pre KDD phase and sampling, also deal with business issues[AS08]. If they did not it would not be possible to extract a meaningful sample. [AS08] The correspondences identified are presented in 2.1:

Table 2.1: KDD processes correspondence summary

KDD	**SEMMA**	**CRISP-DM**
Pre KDD	————	Business Understanding
Selection	Sample	Data Understanding
Preprocesing	Explore	
Transformation	Modify	Data Preparation
Data Mining	Model	Modeling
Interpretation/Evaluation	Assess	Evaluation
Post KDD	————	Deployment

2.3.2 Related projects

The following sections present and explore some projects related to the one in hand.

2.3.2.1 Using hierarchical clustering algorithms for the Turkish residential market

[HV11], studied the contributions of clustering methods in the assessment of real estate portfolios. The study applied a set of hierarchical clustering techniques (average, centroid, complete, median, single, ward and weighted), to a dataset containing rental return data from seventy-one metropolitan residential areas in Turkey.

Clustering techniques are able to describe a dataset by partitioning it into groups, where the instances belonging to the same group are more similar among themselves than data from different groups. Hierarchical clustering is one of the most widely adopted clustering methods as it allows for a simple and effective visual representation of the results while also allowing for general usage since the user does not have to input parameters for the algorithm. Commonly, a tree like structure called dendrogram is used to represent each step of the agglomerating process. By cutting the tree at different levels clustering results are obtained. The tree obtained in this project is presented in figure 2.4

Figure 2.4: Clustering results dendrogram. [HV11]

The results of this study support the partition of residential areas in three clusters that reveals a significant return profile distinction in the residential areas in Turkey. This made it possible to identify a group which was specified as the "hot" market for real estate investment, with clearly higher return rates than others.

2.3.2.2 A Neural Network based model for real estate price estimation considering environmental quality of property location

In this paper [CCMO14], the authors propose a model based on ANN, artificial neural network, for real estate price estimation. ANNs are among the best methods for learning input-output relationships from observable data. In a complex system like the real estate market this ability may prove very useful to model the market's trends and motivations.

The study collected data regarding asking prices of properties in Taranto, Italy, to serve as training data for the task of estimating the asking prices of residential assets. For this the authors studied the impact of different variables: environmental, property intrinsic, distance to city centers and pollution levels.

The ANN proved very efficient for the task proposed. The authors were able to create a model with a very good correlation to both train and test results. They also performed a sensitivity analysis in order to assess the most significant input variables. This was achieved by repeating the ANN training phase and removing one feature at each iteration while evaluating the impact of the removal. As a result of this analysis the variables which seem to be the most significant are those related to the address of the asset. The addresses with proximity to industrial centers, with the existence of a garden, balcony, or proximity to the beach were most significant. The number of inhabitants in the area does not seem to affect property values as its removal from the dataset resulted in no changes to the model.

This model may be used to support investment in the transportation areas, help with appraisals and possibly even promote a positive environmental impact.

2.3.2.3 Visualization-Aided Exploration of the Real Estate Data

[LBSY16], implemented a visualization-aided system for buyers to explore the real estate data of entire cities. Using this system for different areas it is possible to explore school and transportation systems as well as other facilities. It also allows the user to filter by individual requirements and make comparisons between properties/suburbs. They provide a preliminary implementation of the system where it is possible to explore all these features.

The real estate data was collected by non-conventional channels such as crawling information from Real Estate Australia, census statistics and transportation companies' websites. The data from different categories was integrated and transformed into attributes. Finally, they propose a visualization solution with four coordinated views:

- Google map view - Maps the real estate assets with geometrical figures on a google map view and uses visual variables and colours to provide additional information.
- Multidimensional view - The multidimensional view uses the current map view to connect parallel coordinates with the attributes, to a geographic scatter plot and a coloured boolean table. This view is very effective as it can model the entire map view or just the property the mouse points at. It also allows for effective visualization of different sets of property features.
- Image card view - Displaying images of the neighbourhood.
- World cloud view - Is a visualization technique of textual information where the importance of each tag is reflected on the font size.

In figure 2.5 it is possible to see a preview of the platform exploring region prices in a visual way.

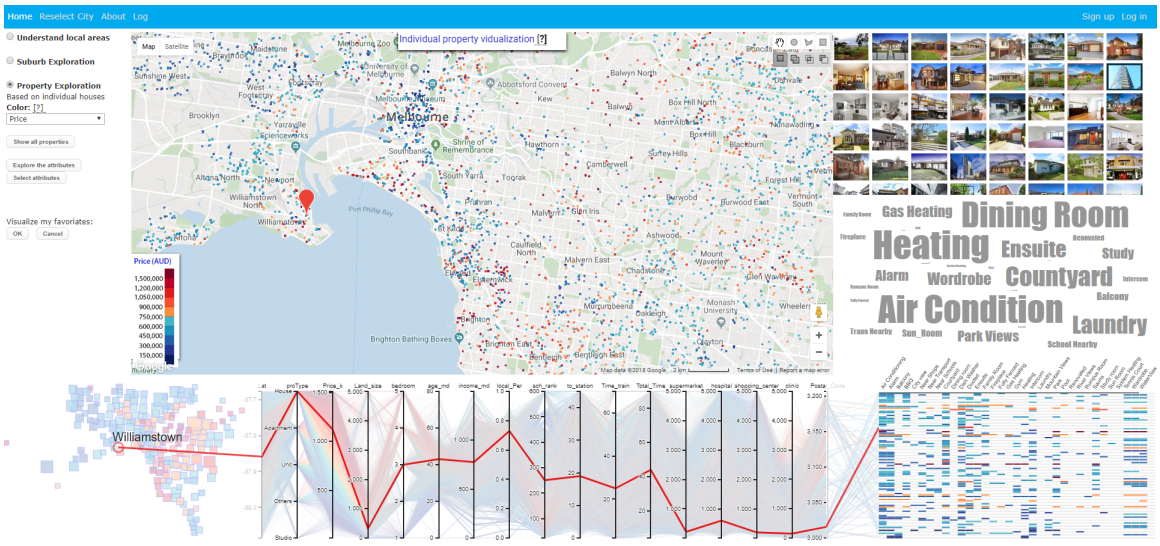

Figure 2.5: Visual aided real estate exploration platform

2.4 Conclusions

Having reviewed the state of the art concerning information retrieval methods, web scraping in particular, it is concluded that this methodology appears to be effective and efficient for the purposes of this project. By making use of Beautiful soup 2.2.1.1 and Selenium 2.2.1.2, it will be possible to retrieve the contents of the target real estate platforms for storing, exploring and constructing our dataset. It is important to state that web scraping may be against the terms of use of some websites and platforms. Being that in this project our interest is on the scientific issues that concern the adoption of scraping mechanisms we do not take legal issues into account when implementing the scrapers and Web scraping techniques.

The DM field appears to be in the prime of its existence, its applications are numerous and in very distinct areas. In the real estate market the importance of delivering pertinent, useful and insightful information regarding assets has led to the study of data mining applications for real estate appraisal. This has also led to the development of more interactive and valuable visualization techniques. Environmental factors also seem to play a major role in real estate pricing motivations and trends, and is closely followed by the intrinsic properties of the assets. The latter, is the target data for this project.

Chapter 3

Scraping Application

3.1 Introduction

This chapter describes the implementation of the web scraping application for collecting real estate data. The application targets the data in eight selected web platforms for real estate advertising in Portugal. The scope of the project is limited to data collected in the Porto district, as it is sufficient to validate all aspects of the proposed solution. The main requirements of the system being developed are the continuous collection, storage and update of the retrieved data. The purpose is to not only collect the available data at the moment but to build a solution which enables the monitoring of the selected platforms. This allows the platforms to build up the data stored over time and to detect changes in the assets' prices.

The following sections present the main findings of the studies performed on the real estate platforms selected, the specifications of our database and its contents, and the description and presentation of the system's architecture. The chapter ends with a discussion about the challenges faced throughout the development, how they were overcome and the main limitations of the built system.

3.2 Data sources

The first step towards planning and building the proposed application was to explore the web platforms operating in the Portuguese real estate market. Several platforms were identified in the first selection phase with a focus on the Porto district and its data regarding location, prices, typologies, photos and time-stamps. Table, 3.1 portrays the absence of the desired data on the explored platforms:

The most valuable data for the goals and tasks of our project is in the form of time-stamps, location details, prices and typologies. As exhibited by the results, all the explored platforms offer information about location, prices and typologies. So our selection criteria for this stage was

Table 3.1: Absense of desired data from the explored real estate platforms

Platform	Imovirtual	Idealista	Era	Remax	Casas Sapo	OLX	CustoJusto	BPIExpresso	MilleniumBCP	Caixa	LardoceLar
Price											
Photos											
Timestamp			X	X					X	X	
Useful area										X	
Terrain area		X							X	X	X
Location											
Features		X		X	X	X	X		X		X
Textual description			X								
Typology											
Contacts			X		X				X		X
Purpose											
Views			X		X		X	X	X	X	X
Platform ID											

based at first, on the availability of time-stamps and then two more platforms were included for its abundance of data. Given these points, nine scrapers were developed for the continuous retrieval, update and storage of the desired data on the following platforms:

- Imovirtual - https://www.imovirtual.com
- Idealista - https://www.idealista.pt
- Casas Sapo - https://casa.sapo.pt
- OLX - https://www.olx.pt/imoveis
- CustoJusto - https://www.custojusto.pt/portugal/imobiliario
- BPI Expresso - http://bpiexpressoimobiliario.pt
- LarDoceLar - https://www.lardocelar.pt
- Remax - https://www.remax.pt
- Era - https://www.era.pt

Finally, we explored the structure and organization of the platforms. Our findings support that we should guide our implementation by the two major tasks identified: scraping advertisement links and scraping those links' contents. The identification of an asset in a given platform is achieved by means of a platform ID. Therefore, in our system the assets are identified by the pair <platform name, platform ID>. If the platform does not provide an ID for an asset then we aggregate the title, price and type of asset into a string to serve as an alternate ID.

3.3 Data storage

For storing our data we use a MongoDB database. Mongo is a NoSQL database program which uses documents similar to JSONs to represent instances. It is a great fit for the task in hand as it allows us to store data in a flexible structure. Mongo offers all the main features of regular

database systems and more: powerful queries, indexing, replication, load balancing, file storage, aggregation, server-side Javascript execution, fix-sized collections and transactions. [1]

After a careful analysis of the available data, a model was defined for the advertisements being stored. It contains all the features that were considered relevant for the current project as well as some others that may be used for further enhancing the functionality of the proposed system. Table 3.2 presents the documents' fields names, types and a short description.

Table 3.2: Advertisements storage model

Name	Type	Description
_id	objectID	internal mongo identifier
photos	arr(string)	array of photo links
price	float	asset's asking value
typology	string	real estate nomenclature for housing assets. (T0 - TX, X stands for number of bedrooms)
usefulArea	float	constructed area
terrainArea	float	construction plus terrain area if applicable
otherCharacteristics	arr(string)	array of other characteristics found (construction year, condition, ...)
features	arr(string)	array of asset's features (fireplace, parking spot, balcony, ...)
description	string	a textual description of the conditions advertised
district	string	asset's district
municipality	string	asset's municipality
parish	string	asset's parish
platformID	string	platform unique identifier
views	int	number of visualizations of the advertisement
creationDate	date	timestamp of the advertisment's creation
updateDate	date	timestamp of the advertisment's last update
contact	int	phone number to contact the announcer
purpose	enum()	intention to sell ou rent
platform	enum()	source platform
type	enum()	type of asset being advertised: apartment, house, bedroom, terrain, store, warehouse, garage, office, building, farm or loft
link	string	URL of the scraped advertisement
updates	arr(object)	each object of the array represents a detected update and contains some data on it
object.oldPrice	float	asset's price before the update
object.newPrice	float	asset's price after the update
object.Timestamp	date	timestamp of the change detection
object.newLink	string	URL of the advertisement containing the updated price

3.4 System architecture

The system built follows the application design pattern known as Producer/Consumer. The main principle is that there is one or more threads that produce some desired instance that is stored in a queue and properly handled by one or more consumer threads. This design pattern is best applied when the rate of production is significantly higher than the consumption. There are several components working together for the purposes of the application. A detailed architecture diagram is presented in figure 3.1.

- Queue - A queue is an abstract data structure that stores its contents by order of arrival. The operations allowed in this structure are restricted to inserting elements on its tail and

[1] This and more information about mongodb on the website https://www.mongodb.com/

removing from the first position. For our solution, we implement a different queue for every platform and it serves as a communication channel for our producer and consumer threads.

- Link producer - The producer threads are responsible for retrieving advertisement links and placing them in the corresponding queue. There is one instance per platform. It accesses a given URL and loops through the pages of results effectively sending links to the queues. The thread terminates after the last page of results is scraped.

- Link consumer - The consumer threads continuously access the queues and remove the top element until both the queue is empty and the producer thread has signaled that there are no more links. It is possible to configure the number of consumer threads desired. Each thread sends a request to the URL removed from the queue and scrapes its contents while structuring them in the model presented in table 3.2. This structure is then stored as a document in our Mongo collection. This thread also manages the updates of existing advertisements. If a already exists in our database (detection via ID) the price of the current advertisement is checked against the one that exists in our database. If a price change is detected a new field is added to our asset's document: the updates field. It consists of an array of updates where each instance contains some information related to the changes.

- Main - The main program for our solution is responsible for all the previously presented parts working together. It takes the target platform as single argument, establishes the connection to the database and launches the producing and consuming threads. When the producer is no longer retrieving links it signals the consumers so that when the queues are empty they no longer wait for more links and all the threads can safely terminate.

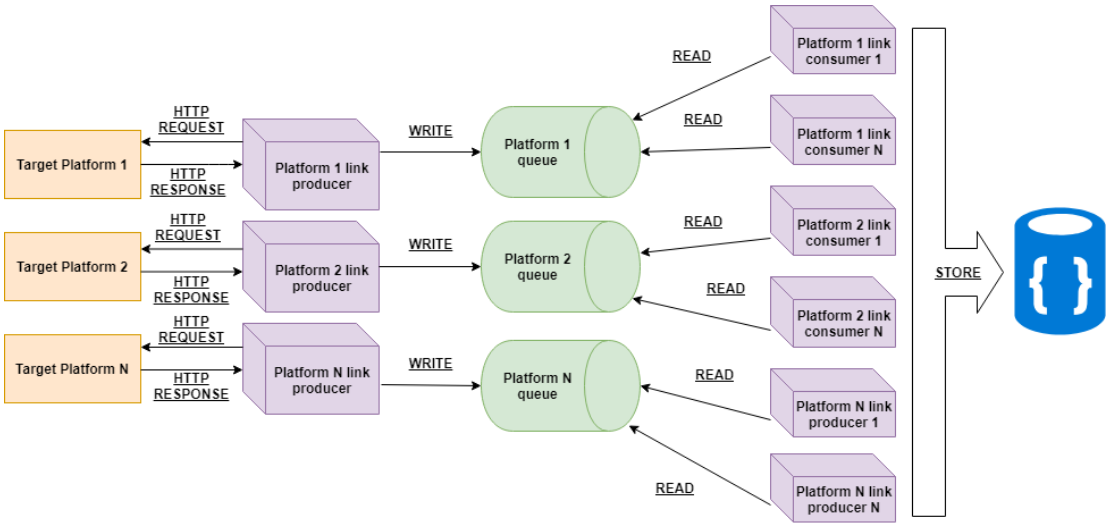

Figure 3.1: Scraping application architecture diagram

3.5 Collected data

The creation of the described scraping application enabled the collection of a wide range of advertisements from the target real estate platforms. Its continuous usage throughout the last months also contributed to a good amount of registered price updates on our database. Table 3.3 presents a broad view of the total amounts of scraped advertisements and their platform distribution.

Table 3.3: Collected data amounts and distribution by platform

Platform	**Number of assets**	**Number of updates**
Imovirtual	96993	3954
Casas Sapo	29099	2095
OLX	5625	491
CustoJusto	50183	2228
BPIExpresso	28013	1881
LarDoceLar	12804	606
Remax	3122	150
Era	11624	386
Total	237463	11791

3.6 Challenges and limitations

There were several challenges during the development of this application. Most of them were connected to some active target protection against scraping and non-human usage of their platforms, while others were related to the nature of the data being collected.

The first problem encountered was that after a number of GET requests were sent, some platforms would actively block the IP address of the source. To tackle and mitigate this issue as much as possible we developed some "anti-detection" mechanisms. A GET request contains a singular string that allows the identification of the source operating system, application type and software versions by network peers: the *user-agent* field. Using data from a web publication about user-agent's percentage usage, a random choice function was implemented. It chooses from a list of the most common user agent where each element has a probability of being chosen associated to their predominance. This technique is employed on both the Link producer and the Link consumer whenever a GET request is sent. In-between every request sent to the platforms there is a random waiting time between one and three seconds in both threads. Finally, when the Link producer sends the links extracted from the current list page to the queue, there is a randomization of the order in which they are sent, so that the access is not sequentially performed.

Another common issue was page content that would only render after some user action. This means that the desired content is hidden on the initial render but it can be obtained through the automation of the user task needed. When faced with dynamic content we used Selenium, reviewed in 2.2.1.2. It makes use of a web browser driver to automate actions or wait for specific elements to load. The main usage for this was on the *Remax* platform where the pagination was dealt with

Javascript rather than URL parameters. Almost all platforms protect the contact information of the advertisement's owner. It was decided that this work-around would not be used for scraping this data because of the overhead added. Accessing every URL scraped in this manner is an intensively time consuming task and for the scope of the current project, contact information was deemed irrelevant.

Regarding the asset's location, most of the target platforms do not present the full address on their advertisements. Typically, a string aggregates all the available information about the location and there may be a Google map indicating the broader area where it is located. These strings were matched against lists of Porto's municipalities and parishes whenever this information was not distinctively presented.

The performance of these enhancements can be evaluated by their ability to overcome the difficulties in the scraping mechanisms for eight out of nine selected platforms. Regardless of our best efforts, the Idealista platform proved able to actively block our attempts to develop automated mechanisms for accessing its pages even after the implementation of the before mentioned enhancements.

Chapter 4

Price fluctuations classification

4.1 Introduction

In this chapter we describe the classification task that was performed by using the data gathered with the scraping application presented in the last chapter. This data presents an opportunity to perform a prediction task to classify the assets' price fluctuations. With this motivation, *Rapidminer* was chosen to serve as a software platform in which we performed our data mining tasks and procedures. Rapidminer provides an environment with a drag and drop interface that offers a wide range of operators for data preparation, machine learning, deep learning, text mining, and predictive analyses. More information on Rapidminer can be found on their website [1]. Using a NoSQL extension for the Rapidminer environment enabled us to establish a connection between Rapidminer and our local mongo database. Then, using the *Read from Mongo* operator we queried our database to obtain our data in a structured data table. With the tools and operators made available by Rapidminer we set to perform our data mining task.

The following sections begin by presenting and describing the dataset used in this DM task. After that all the preprocessing tasks performed and their methodologies are detailed. This is followed by the description of the predictive methods employed, the different experiments conducted and their results discussion and evaluation. To conclude the chapter the main conclusions and limitations of our task are briefly discussed.

4.2 Dataset description

As shown in table 3.3 our database has a lot of raw data. For the purposes of this task, some of the model's fields were deemed not fit. This was the case of the *otherCharacteristics* and *features* arrays as they represent information that is hard to properly categorize and standardize. The textual description is also not useful without some previous text mining preprocessing technique which is

[1] https://rapidminer.com

not in the scope of this project. We were only able to extract the number of views and contacts from a small number of platforms therefore, these are also not used in this analysis. The platform name, ID and asset photos were also considered not useful for serving as features for this task. Table 4.1 presents the dataset extracted from our database and describes the first exploratory findings about the data that will be used for the classification task. The dataset contains 222933 entries and each represent an asset's advertisement.

Table 4.1: Exploratory findings on extracted dataset

Name	Type	Missing values	Least	Most	Min	Max	Average	Deviation
_id	Polynomial	0	-	-	-	-	-	-
district	Polynomial	0	Porto	porto	-	-	-	-
municipality	Polynomial	24253	vilar do paraiso	cedofeita	-	-	-	-
parish	Polynomial	66805	felgueiras	porto	-	-	-	-
creationDate	Real	66277	-	-	-	-	-	-
price	Real	502	-	-	0	280000000	249748.830	898076.656
purpose	Polynomial	8076	rent	buy	-	-	-	-
usefulArea	Real	37031	-	-	0	3000000	620.402	15100.566
terrainArea	Real	178574	-	-	0	54760000	6827.301	280654.039
type	Polynomial	2777	negocio	apartamento	-	-	-	-
typology	Polynomial	66910	t15	t3	-	-	-	-
updates.newPrice	Real	211231	-	-	0	20000000	248546.368	74188414.024
updates.oldPrice	Real	211305	-	-	1	8000000000	927474.720	15100.566

A careful analysis of the information presented in table 4.1 and in the frequency tables for each attributes' values in section A.1 reveals several problems regarding data quality and consistency. There are a lot of missing values in several attributes therefore, it is necessary to employ proper strategies to deal with them. Also, there seem to be some inconsistent minimum and maximum values for some attributes. The existence of areas and prices with the value zero also point to the existence of entries that we may choose to discard. Properly detecting and dealing with these is also decisive. Finally, problems regarding the consistency of string values were also found. Some values have extra white spaces, capitalized letters and accents and it is necessary to make all this information consistent in order to perform our DM tasks.

4.3 Data preprocessing

For properly performing our classification task there is a set of preprocessing operations that needs to be done. Regarding data quality these are dealing with missing values, properly detecting outliers, removing noisy data and dealing with some other values' inconsistencies. The following sections begin by presenting the strategies involved in solving these quality issues. After that, we present the label of the training and test instances and explore our resultant dataset. Finally, we present and describe how the clustering and feature engineering were part of our preprocessing.

4.3.1 Data quality

The quality of the models and results produced in DM tasks is closely related to the quality of the data used. Quality issues may arise from human-errors, data collection and integration. These

may result in datasets with duplicate, inconsistent and noisy data. Using this data without previously dealing with these issues could result in models that do not portray the true patterns in the data. Removing or reducing these problems should result in an improvement of the quality of the knowledge extracted.

4.3.1.1 Missing values

As shown in table 4.1 there is a large amount of missing values in our dataset. As expected the *_id* and *district* fields do not have missing values. As for the other fields, we adopted different strategies for handling the replacement of the absent value. On the *district*, *municipality*, *parish*, *typology* and *type* fields, the missing value was replaced for an "unknown" string. On the *price*, *usefulArea* and *terrainArea* the value was replaced by the average of the aggregation between *type*, *typology* and *purpose*. For the *purpose* the rows with missing values were discarded for the lack of means to properly predict the correct value. No strategy was adopted regarding the replacement of missing values on the updates' data as their absence was expected due to the nature of the variable.

4.3.1.2 Data consistency

As mentioned in section 4.2 several issues regarding data consistency were identified. Several steps were followed to solve the inconsistencies found. The first was to transform all the values in the dataset to lowercase. After that, using the *Trim* operator all the white-spaces occurring in the beginning or end of a string were removed. The next step was to replace the occurrences of all possible combinations of vowels with accents in the Portuguese language by the vowel alone. Finally, some inconsistencies regarding value types were also found present in the dataset. As they were present in a very small number of entries, the solution employed was the removal of these rows from the dataset.

4.3.1.3 Outlier Detection

In a dataset, outliers are anomalous objects or values. The presence of outliers in a dataset may indicate the existence of noise. To try and mitigate both these issues, an outlier detection task was performed. This task was performed on Rapidminer using its operator *Detect outliers*. Timewise, the performance was very poor: a single run takes hours. To tackle this we generated ten random samples of the dataset each containing a tenth of the original set. We performed the outlier detection on these samples taking into account the five closest neighbours, according to the *Euclidean distance* and classifying two outliers per sample. The samples are then aggregated and the rows marked as outliers are removed.

4.3.2 Dataset labeling

In order to properly perform a classification task we need to create a model that classifies the instances of our assets into predefined categories. Labeled data is needed in order to properly train

and test the built model. The information stored in our database regarding the updates of assets' prices is determinant for the success of our DM task.

The first step was to extract a dataset in which every instance suffered an update. After the preprocessing detailed in the previous sections we obtained the dataset used for the labeling.

Using this set and we managed to calculate the value of the difference of prices after and before the updates. When the difference is positive the instance is labeled as "up" when it is negative it is labeled as "down". Table 4.2 presents the obtained labeled dataset and figure 4.1 presents the distribution of assets by class. The dataset contains 11486 rows.

Table 4.2: Description of the labeled dataset

Name	Type	Missing values	Least	Most	Min	Max	Average	Deviation
_id	Polynomial	0	-	-	-	-	-	-
district	Polynomial	0	porto	porto	-	-	-	-
municipality	Polynomial	0	marco canaveses	porto	-	-	-	-
parish	Polynomial	0	vilar do torno e alentem	cedofeita	-	-	-	-
creationDate	Real	2957	-	-	-	-	-	-
price	Real	0	-	-	-0.451	35.820	0.000	1.001
purpose	Polynomial	0	rent	buy	-	-	-	-
usefulArea	Real	0	-	-	-0.100	82.251	0.000	1.001
terrainArea	Real	0	-	-	-0.229	89.144	0.000	1.001
type	Polynomial	0	negocio	apartamento	-	-	-	-
typology	Polynomial	0	t5+1	t3	-	-	-	-
updates.newPrice	Real	0	-	-	0	20000000	248500	551821.928
updates.oldPrice	Real	0	-	-	1	1700000	239522	545000.137
class	Binomial	0	up	down	-	-	-	-

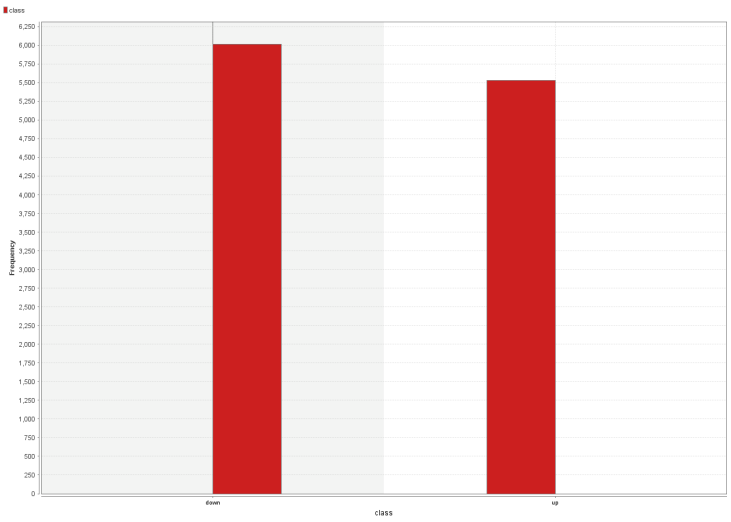

Figure 4.1: Classes' distribution

4.3.3 Clustering

For the purpose of better understanding our dataset we performed a clustering task. As previously described in section 2.3.1.1 clustering techniques refer to the positioning of items in a finite number of sets where the set's items are more similar to each other than to data from other groups.

With this goal we employed different clustering techniques to obtain the best possible results:

- DBSCAN - Our first clustering attempt made use of Density-based spatial clustering of applications with noise (DBSCAN). It is a density based-based algorithm and it creates clusters from the zones of higher density. Objects that do not fit into theses dense zones are considered as noise. Despite many attempts considering different numbers of neighbours and measure types, the results of the DBSCAN were never useful. There was always a cluster with almost all the entries while the others assumed residual counts. An example of the outputs obtained is presented in figure 4.2

```
Cluster 0: 8506 items
Cluster 1: 5 items
Cluster 2: 5 items
Cluster 3: 8 items
Cluster 4: 5 items
Total number of items: 8529
```

Figure 4.2: DBSCAN results

- K-Means - This is one of the most popular algorithms for clustering and it is classified as a prototype-based method. The aim of this technique is to place entries in the group with the closest mean to their own. Initially each entry is a prototype of its own cluster. Over the iterations the solution converges to a predefined number of partitions. One of the main disadvantages of using this algorithm is the necessity to define the number of clusters beforehand. An approach to doing so is to plot the *within-groups sum of squares* for different numbers of partitions and obtain what is referred to as the *elbow curve*. Figure 4.3 shows the result from plotting the aforementioned curve.

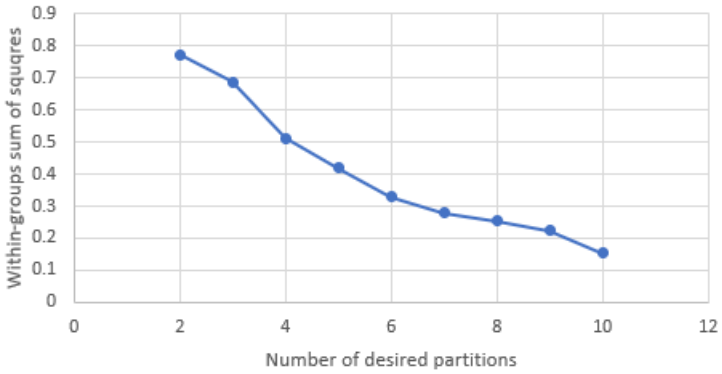

Figure 4.3: Elbow curve

The within-groups sum of squares is an inner cluster distance measure. In order to balance compactness and utility a value from the elbow zone of the graphic is chosen as the number

of partitions for the clustering task. In our case six was chosen. The results obtained with this method are much more promising. One major cluster was identified, size-wise. There is a second cluster with a relatively big size and the remaining entries are evenly distributed on the other clusters. The results of this task are presented in figure 4.4. Finally, a new attribute was added to every instance. It represents the cluster the entry belongs to.

```
Cluster 0: 915 items
Cluster 1: 390 items
Cluster 2: 4465 items
Cluster 3: 436 items
Cluster 4: 1459 items
Cluster 5: 864 items
Total number of items: 8529
```

Figure 4.4: K-MEANS results

4.3.4 Feature engineering

The feature engineering step of the preprocessing process refers to the selection of the attributes from our dataset that will be used as predictive attributes and the generation of new features from the existent attributes.

Further considering the dataset available some attributes were considered not relevant or inadmissible as predictive attributes. The fields *newPrice* and *oldPrice* are exclusive of items belonging to the training set. Therefore, these attributes were discarded. The attribute *creationDate* was also left out as a consequence of the creation of a new temporal attribute: *creationMonth*. This is a calculated value that refers to the month since the *epoch*[2] in which the advertisement was created. Another generated feature was the *pricePerArea*. Its value is defined by the expression $price/usefulArea$. The results of the clustering task described in 4.3.3 were used as predictive attributes in the form of the cluster each instance belongs to. Lastly, two more features that capture the location attributes of the assets were created. For this purpose, the rows with missing values in the *creationDate* field were discarded. By grouping the instances by *district*, *municipality*, *parish*, *creationMonth*, *purpose*, *type* and *typology* we were able to perform an aggregation by price average. We also counted the number of assets for each of the groupings. These values were then appended to the assets of the respective groupings.

All this considered, the set of predictive attributes chosen for the classification tasks is presented in table 4.3:

[2]The Unix epoch (or Unix time or POSIX time or Unix timestamp) is the number of seconds that have elapsed since January 1, 1970 (midnight UTC/GMT)

Table 4.3: Predictive attributes

Predictive attributes
district
municipality
parish
price
type
typology
purpose
usefulArea
terrainArea
cluster
averagePricePerGrouping
countPerGrouping
creationMonth
pricePerArea
class

4.4 Classification

A classification task is a predictive task that aims to label new, unlabeled objects. This label represents a class or a category and its given by the objects' predictive attributes. The applications of classification tasks are numerous and very diversified. From classifying your email as spam, to distinguishing fraudulent transactions from regular ones all private and public sectors employ classification in many activities.

The classification task performed is a case of binary classification. Using the labeled dataset shown in table 4.2, two different algorithms were used in the training process:

- Random Forest - The random forest method builds several decision trees and combines them to obtain a more accurate and stable prediction. In decision trees as we get into deeper levels there is a tendency for overfitting. This algorithm attempts to prevent this by creating random subsets of trees and building smaller trees from these.

- Single Layer Perceptron - As mentioned in section 2.3.2.2 ANN are mathematical models that are very effective at learning relationships between inputs and outputs. The perceptron is a type of ANN with a single layer of neurons. It is commonly classified as the simplest feed-forward neural network: a linear classifier. A feed-forward network is an ANN where the information only moves in one direction, from the input nodes to the output nodes. The weights are increased or decreased according to the correctness of the output and the chosen learning rate.

4.4.1 Results and experiments

The following sections present the different experiments carried out for our classification task and their results.

By training models with the before mentioned algorithms, we were able to classify our asset's in the desired classes, *up* and *down*. After this the results were analyzed and evaluated.

In both experiments the results are evaluated by means of a *cross-validation* operator. This operator has two subprocesses: training and testing. The model is trained in the training subprocess and the trained model is then applied to the testing subprocess. It is also in the testing phase that the performance of the model is evaluated. *Cross-validation* works by partitioning the input dateset into folds, five in our case. Of these five a single one is chosen as a test set. This validation process is repeated so that each set is used as a test set exactly once. The results are then averaged to produce a single *confusion matrix* and an accuracy measure.

4.4.1.1 Experiment 1 - Random Forest

- Parameters- The chosen training algorithm for this experiment expects four arguments: number of trees to build, number of features to consider, seed value for random generation and the maximum depth of the trees. In the present experiment the best results were obtained by building two hundred trees while considering four random features. The seed value was set to one and the depth indicated as unlimited.

- Confusion matrix

Table 4.4: Confusion matrix for experiment 1

	true up	**true down**	**class precision**
predicted up	2574	1150	69.12%
predicted down	1447	3358	69.89%
class recall	64.01%	74.49%	

- Accuracy: 69.55% +/- 0.60%

4.4.1.2 Experiment 2 - Single Layer Perceptron

- Parameters- The single layer perceptron takes two arguments: number of weight updating rounds and the learning rate that determines how much the weights will be modified. For this experiment the best results were obtained with 25 rounds and a learning rate of 0,03.

- Confusion matrix

- Accuracy: 52.87% +/- 0.86%

Table 4.5: Confusion matrix for experiment 2

	true up	**true down**	**class precision**
predicted up	1528	1527	50.02%
predicted down	2493	2981	54.46%
class recall	38.00%	66.13%	

4.5 Conclusions and limitations

Having concluded our classification task with moderate success appears to indicate that the methodology employed was adequate for the problem at hand.

In terms of results evaluation, the model built with the random forest algorithm shows significantly better performance than the single-layer perceptron's. This is probably due to the algorithm's capability to withstand scaling, feature transformation and irrelevant feature inclusion. In this experiment, as shown in table 4.4, the precision and recall are closely related. On the other hand, ANN and specifically single-layer perceptrons may need additional preprocessing of the data or even require the use of a more complex ANN in order to produce comparable results.

The training set used for this task was collected in a relatively small window of time. Our classification results could improve given more time to collect more price updates as the training set would be larger. Another constraining aspect of our task's success is the lack of location-specific data like census data which could prove effective and valuable when trying to model the real estate asset's prices fluctuations.

Chapter 5

Web platform

5.1 Introduction

In order to provide listing and searching mechanisms for the findings described in the previous chapters we set out to build a web platform. This platform features basic and advanced search by different criteria, ability to list and preview asset's information as well as access to local and temporal price averages and fluctuations as well as the results of the classification task described in chapter 4, for the particular assets.

To better describe the implementation and deployment of the platform the following sections will begin by distinguishing and presenting the *front-end* and the *back-end* and their implementation methodologies. This is followed by the demonstration and explanation of the broader system's architecture and presentation of the platform's views. Lastly, the limitations and future work are succinctly discussed.

5.2 Technologies

The main technologies used are described in the following sections, together with the main reasons for their selection.

5.2.1 Front-end

For the front-end development of our platform the main requirement refers to the system's availability for web and mobile access. Aiming for a swift, well-structured development and a fit for all solution, *ReactJS* was chosen as the development framework.

React is a *Javascript* framework for building user interfaces originally developed and maintained by *Facebook*. The main advantages of using this technology are related to implementation of the User Interface (UI) in different components. React continuously monitors the Document Object Model (DOM), re-rendering it when some condition is changed. The components are the

changed parts of the DOM in this operation. According to some authors this paradigm could be the future of UI development.

For the purpose of creating a standard, familiar and appealing interface the principles of Google's *Material Design* were applied throughout the development. Material Design consists in a set of design principles that guide UI development. These principles relapse upon motion, depth, light fundamentals, content hierarchy and many other UI aspects. To achieve this Material-UI was chosen. Material-UI is an open-source library that provides React components that implement Google's *Material Design*.

5.2.2 Back-end

A server was implemented for the back-end purposes of our application: providing routes, handling requests and querying the Mongodb database described in section 3.3.

In order to keep our server implementation clean and well-structured, *Express* was used as the development framework. *Express* is a lightweight *NodeJS* framework for web development that structures the *server-side* of the application into a Model View Controller (MVC) architecture.

Joi (NPM package) was used to validate the parameters that can be passed to the route handlers. This package allows us to specify a schema for the *body*, *parameters* and *query* of the requests. When a request is received, the *Celebrate middleware* verifies that the incoming parameters are in accordance with the specifications and immediately raises an error otherwise. The usage of these packages assures that when the handler function receives data, it is properly sanitized.

5.3 System architecture

The implementation of this platform follows a *Client-Server* architecture. There are two main components in this design: clients and servers. The server delivers and manages most of the resources and services which are requested and consumed by the client. This type of architecture allows for several clients distributed over a network to connect to a single server. In our system, a database is also required to access the data that is transformed and displayed in our platform. A detailed architecture diagram is presented in figure 5.1.

- Server - Our server provides a communication channel between the client and the database. When the client needs access to the information on the database it sends an HTTP Request to the server. This request will then be received on one of the server's endpoints. The endpoints made available are presented in tables 5.1, 5.2 and 5.3. As mentioned in section 5.2.2 *Joi* and *Celebrate* are used to assure the sanity of the URL request parameters.

Table 5.1: Advertisement endpoint description

Endpoint	HTTP Method	URI	Parameters	Content	Media Type
Advertisement	GET	http://../advertisement/	ID	advertisement ID	JSON

Table 5.2: Advertisements endpoint description

Endpoint	HTTP Method	URI	Media Type
Advertisements	GET	http://../advertisements/	JSON

Table 5.3: Advertisements' query parameters

Query Parameters	Content	Media Type
page	The desired page of results	JSON
perPage	Number of results per page	JSON
purpose	Buy or rent	JSON
type	Asset's type (apartment, garage ...)	JSON
district	Asset's district	JSON

The query parameters used for the *advertisements* route are structured in such a way that the data returned to the client always consists in a single page of results. To achieve this when the server queries the database a number of results are skipped equal to the result of the operation $page * perPage$ and the results limited to *perPage* results. To avoid having to make two separate Requests, the total count of assets available for that particular query is appended to the response's *Header*.

- Client - The clients correspond to instances of our *front-end* application. Their job is to allow and process user inputs, perform HTTP Requests and properly handle and display their responses.

- Database - MongoDB database containing all the stored advertisements that were collected through the scraping application described in chapter 3.

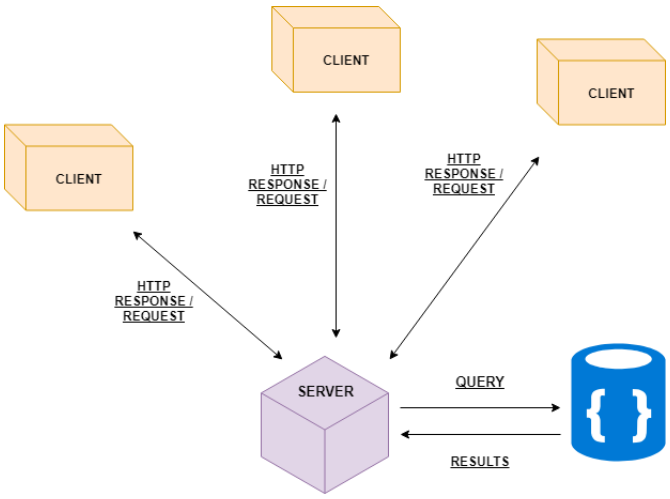

Figure 5.1: Web platform architecture diagram

5.4 Views

The following sections present all the projected views of the platform and their main purposes and features. Some of these views are still not fully functional or lack some aspect of their implementation.

5.4.1 Search view

The search functionality was split into two distinct features: basic and advanced search.

5.4.1.1 Basic search view

The basic search allows for a swift and broad search. It is the view presented in the platform's homepage. To perform a search, the user needs to select two mandatory fields/values and an optional one. The last step is to click on the *SEARCH* button which triggers an *HTTP Request* to the server. Figure 5.2 presents the described view.

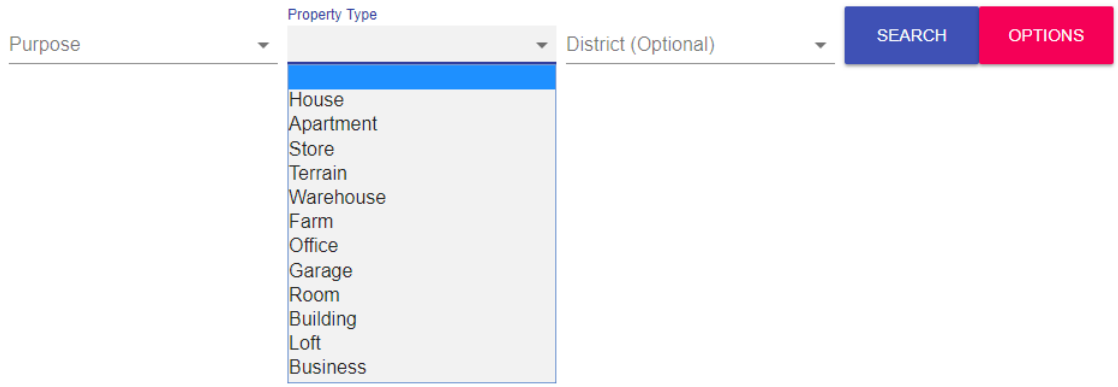

Figure 5.2: Basic search view

5.4.1.2 Advanced search view

The advanced search view offers the possibility to *fine-tune* the parameters of the search query. Clicking the *OPTIONS* button opens a panel with several value and range *select-fields* which the user can adjust to its personal preferences. The described view is shown in figure 5.3.

5.4.2 Results view

The results view is composed of two different main aspects: the listing and previewing of the assets and the pagination of the results.

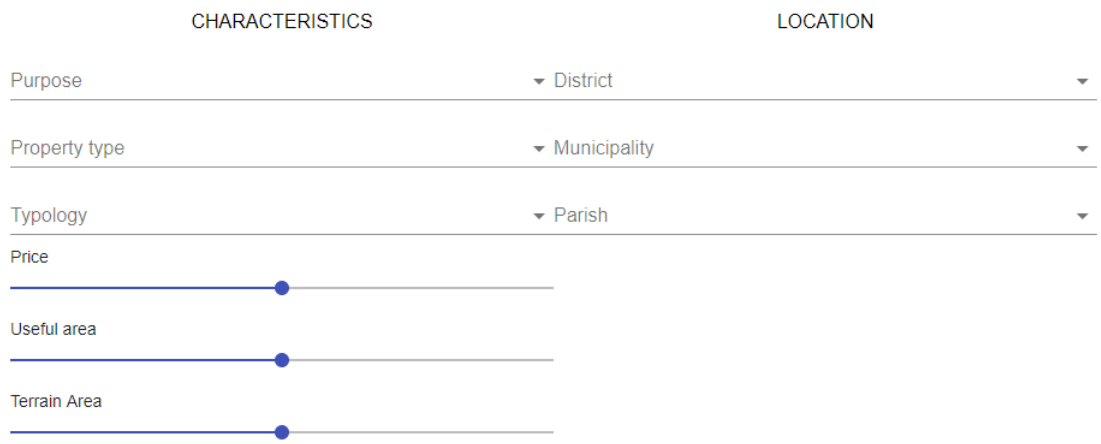

Figure 5.3: Advanced search view

5.4.2.1 Listing view

Each result is presented in its own card in a table row containing the default number of results per page. The card shows a preview of the first available photo, several asset's characteristics and creation date along with the asset's district, municipality and parish. A possible list of results is presented in figure 5.4.

Figure 5.4: Listing view

5.4.2.2 Pagination view

In the last row of the results table there are several controls on how to limit and control the pagination of the query results. There is a select-field for choosing the desired number of results per page, there is a shown range representing the current range of values on the total number of results and four action arrows. The double arrows allow to quickly navigate between the first and last page of results while the others are incremental steps in the above mentioned directions. Figure 5.5 presents the described view.

Figure 5.5: Pagination view

5.4.3 Price analysis view

The price analysis view is accessed through the existing buttons on each entry of the results table. The view is planned to feature four main components:

- Price/Area - Refers to the price by square foot of the current asset. The value is shown together with the deviation from the average square foot price of similar properties in the asset's Parish or Municipality.
- Price averages - Presents current asset's price versus the average price of similar properties in the asset's Parish or Municipality.
- Price history - Presents an historic overview of price averages for the properties described in the previous section. It also allows changing between Parish and Municipality values.
- Price fluctuation prediction - Presents the results of the classification task described in chapter 4 with the possible results being UP and *DOWN*.

This view is still lacking a front-end implementation. The required data and calculations have already been performed and stored but the results are still not embedded in the platform.

5.5 Deployment

Both the *back-end* and *front-end* applications were deployed to *Heroku*. Heroku is a service platform on the cloud that enables the deployment and maintenance of applications from different native languages. In order to make use of the web-platform users access the front-end URL and the HTTP Requests are forwarded to the back-end.

A preview version of the platform is accessible through: https://real-estate-plat.herokuapp.com/

5.6 Limitations and future work

Overall, the development of the described platform and the methodology employed allowed the achievement our main goal: provide the users with the means to make better and informed decisions on their real estate's market financial investments.

The platform provides mechanisms for basic and advanced search, listing, paginating and accessing a preview of the asset's intrinsic and geographic available characteristics. Despite the lack of an implemented analysis view, its projected *mock-up* and the availability of the calculations required should allow for a swift and straightforward implementation in the future.

For future growth of the platform, authentication means could be provided to users. This would allow to capture aspects of user navigation, preferences and concerns. This information could be used to improve the user's experience by providing contextualized recommendations of assets that the user is more likely to be interested in. The possibility to save personal notes, listings and query results could also be made available.

Chapter 6

Conclusions and Future Work

The present chapter concludes this dissertation by addressing and highlighting the main contributions of the work developed and providing a framework for possible future work.

6.1 Contributions

The solution implemented and described throughout this dissertation project proved useful and effective for its established purposes. By making use of this solution, people seeking to invest in the real estate market in Porto now have the means to further analyze and explore their possible investment assets.

The built scraping application described in chapter 3, provides access to a unique and centralized database of assets. As time goes by this database will become more and more valuable. The continuous gathering of asset's information will allow more data on asset' price fluctuations, which should allow even more relevant and insightful knowledge to be presented.

In chapter 4, we described the classification task performed for this project. In order to perform classification tasks a labeled dataset is required. Making use of available information on prices fluctuations we were able to perform this labeling task, as described in section 4.3.2. The produced dataset can be used in many different DM tasks by various data scientists.

Finally, the build web platform is a way of spreading the knowledge gathered to as many people as possible, and making it accessible all over the world.

6.2 Future Work

Considering the current state of the work presented some improvements and features could be added in the future to further enhance the value of the solution.

First, the sources of information can be expanded. There are several other real estate platforms operating in the Portuguese real estate market and their information could be a valuable addition.

Also, the collecting mechanisms should be updated to collect asset's data from all of Portugal districts. Throughout the classification task described, some data inconsistencies were found. These could be documented and used to improve the correctness of the data collection process.

The labeling process of the dataset, as described in section 4.3.2, can also undergo improvements. It would be interesting to use some mechanisms to contemplate multiple updates on the same asset's price. As for the classification task itself, more experiments with different predictive algorithms can be performed. Further preprocessing might also be necessary depending on the chosen algorithms.

Future guidelines for the web platform have already been presented in 5.6. They are mostly related to *front-end* improvements that need to be performed so that the platform can be production admissible and to enable users to authenticate and have access to personalized recommendations.

Overall, there are many approaches to the future work of this project. However, in its current state, it already proves very useful for investors as an alternative to the mainstream platforms, most of which do not provide access to useful insights and knowledge as obtained throughout this dissertation.

References

[AS08] Ana Isabel Rojão Lourenço Azevedo and Manuel Filipe Santos. Kdd, semma and crisp-dm: a parallel overview. *IADS-DM*, 2008.

[CCMO14] Vincenza Chiarazzo, Leonardo Caggiani, Mario Marinelli, and Michele Ottomanelli. A neural network based model for real estate price estimation considering environmental quality of property location. *Transportation Research Procedia*, 3:810 – 817, 2014. 17th Meeting of the EURO Working Group on Transportation, EWGT2014, 2-4 July 2014, Sevilla, Spain.

[Cho10] Gobinda G Chowdhury. *Introduction to modern information retrieval*. Facet publishing, 2010.

[dVDJVDV08] Jackie C de Vries, Hans De Jonge, and Theo JM Van Der Voordt. Impact of real estate interventions on organisational performance. *Journal of Corporate Real Estate*, 10(3):208–223, 2008.

[FPSS96] Usama Fayyad, Gregory Piatetsky-Shapiro, and Padhraic Smyth. From data mining to knowledge discovery in databases. *AI magazine*, 17(3):37, 1996.

[GK03] Gaylon E Greer and Phillip T Kolbe. *Investment analysis for real estate decisions*, volume 1. Dearborn Real Estate, 2003.

[GMCE01] David M Geltner, Norman G Miller, Jim Clayton, and Piet Eichholtz. *Commercial real estate analysis and investments*, volume 1. South-western Cincinnati, OH, 2001.

[Hir14] Jeffrey Kenneth Hirschey. Symbiotic relationships: Pragmatic acceptance of data scraping cyberlaw. *Berkeley Technology Law Journal*, 29:897, 2014.

[HKD13] Abdelhakim Herrouz, Chabane Khentout, and Mahieddine Djoudi. Overview of web content mining tools. *CoRR*, abs/1307.1024, 2013.

[HLA15] Badr Ait Hammou, Ayoub Ait Lahcen, and Driss Aboutajdine. Predicting housing price in moroccan cities using web data. *e-AGE 2015*, page 1, 2015.

[HSK06] Jenny Harding, M Shahbaz, and A Kusiak. Data mining in manufacturing: A review. 128, 11 2006.

[HV11] Ali Hepsen and Metin Vatansever. Using hierarchical clustering algorithms for turkish residential market. 2011.

[Kan11] Mehmed Kantardzic. Data-mining concepts. *Data Mining: Concepts, Models, Methods, and Algorithms, Second Edition*, pages 1–25, 2011.

REFERENCES

[KM06] Lukasz A. Kurgan and Petr Musilek. A survey of knowledge discovery and data mining process models. *Knowl. Eng. Rev.*, 21(1):1–24, March 2006.

[KT00] Mei Kobayashi and Koichi Takeda. Information retrieval on the web. *ACM Comput. Surv.*, 32(2):144–173, June 2000.

[LBSY16] Mingzhao Li, Zhifeng Bao, Timos Sellis, and Shi Yan. Visualization-aided exploration of the real estate data. In Muhammad Aamir Cheema, Wenjie Zhang, and Lijun Chang, editors, *Databases Theory and Applications*, pages 435–439, Cham, 2016. Springer International Publishing.

[NW17] Erik Lee Nylen and Pascal Wallisch. Chapter 10 - web scraping. In Erik Lee Nylen and Pascal Wallisch, editors, *Neural Data Science*, pages 277 – 288. Academic Press, 2017.

[RS70] Richard U Ratcliff and Bernard Schwab. Contemporary decision theory and real estate investment. *Appraisal Journal*, 38(2):165–187, 1970.

[SA05] Manuel Filipe Santos and Carla Sousa Azevedo. *Preâmbulo [a]" Data mining: descoberta de conhecimento em bases de dados"*. FCA-Editora de Informática, Lda, 2005.

[Sch12] Michael Schrenk. *Webbots, spiders, and screen scrapers: A guide to developing Internet agents with PHP/CURL*. No Starch Press, 2012.

[SGW+95] Ted E Senator, Henry G Goldberg, Jerry Wooton, Matthew A Cottini, AF Umar Khan, Christina D Klinger, Winston M Llamas, Michael P Marrone, and Raphael WH Wong. Financial crimes enforcement network ai system (fais) identifying potential money laundering from reports of large cash transactions. *AI magazine*, 16(4):21, 1995.

[VU12] Eloisa Vargiu and Mirko Urru. Exploiting web scraping in a collaborative filtering-based approach to web advertising. *Artificial Intelligence Research*, 2(1):44, 2012.

[Wei05] Gary M. Weiss. *Data Mining in Telecommunications*, pages 1189–1201. Springer US, Boston, MA, 2005.

[WH00] Rüdiger Wirth and Jochen Hipp. Crisp-dm: Towards a standard process model for data mining. In *Proceedings of the 4th international conference on the practical applications of knowledge discovery and data mining*, pages 29–39. Citeseer, 2000.

[YLKK13] Xiaofang Yuan, Ji-Hyun Lee, Sun-Joong Kim, and Yoon-Hyun Kim. Toward a user-oriented recommendation system for real estate websites. *Information Systems*, 38(2):231 – 243, 2013.

[Zim10] Michael Zimmer. “but the data is already public”: on the ethics of research in facebook. *Ethics and Information Technology*, 12(4):313–325, Dec 2010.

Appendix A

Classification

A.1 Exploratory findings

Index	Nominal value	Absolute count	Fraction
1	apartamento	99088	0.450
2	moradia	53648	0.244
3	terreno	26891	0.122
4	loja	19538	0.089
5	escritório	5260	0.024
6	garagem	3636	0.017
7	predio	3132	0.014
8	armazém	3003	0.014
9	armazem	2607	0.012
10	quinta/herdade	1590	0.007
11	quarto	950	0.004
12	quinta / herdade	462	0.002
13	escritorio	282	0.001
14	loft	57	0.000
15	negócio	12	0.000

Figure A.1: Frequency table of Type attribute

Index	Nominal value	Absolute count	Fraction
1	buy	202203	0.941
2	rent	12132	0.056

Figure A.2: Frequency table of Purpose attribute

Index	Nominal value	Absolute count	Fraction
1	cedofeita	11673	0.075
2	paranhos	7500	0.048
3	mafamude e vilar do para...	6548	0.042
4	rio tinto	5932	0.038
5	bonfim	5843	0.037
6	matosinhos e leça da pal...	5574	0.036
7	lordelo do ouro e massar...	5129	0.033
8	canidelo	4951	0.032
9	santa marinha e são ped...	4595	0.029
10	ramalde	3952	0.025
11	cidade da maia	3902	0.025
12	campanhã	3708	0.024
13	gulpilhares e valadares	3525	0.023
14	são mamede de infesta e...	3441	0.022
15	águas santas	3172	0.020
16	fânzeres e são pedro da ...	3158	0.020
17	ermesinde	3057	0.020
18	oliveira do douro	2179	0.014
19	arcozelo	1878	0.012
20	pedroso e seixezelo	1617	0.010

Figure A.3: Frequency table 1 of Parish attribute

Index	Nominal value	Absolute count	Fraction
78	custóias, leça do balio e ...	246	0.002
79	Árvore	243	0.002
80	lordelo	240	0.002
81	baguim do monte (rio tinto)	238	0.002
82	bem viver	233	0.001
83	freamunde	225	0.001
84	penhalonga e paços de g...	223	0.001
85	azurara	221	0.001
86	frazão arreigada	219	0.001
87	soalhães	219	0.001
88	São Mamede de Infesta e...	217	0.001
89	Ermesinde	215	0.001
90	perafita, lavra e santa cru...	214	0.001
91	Cedofeita, Santo Ildefons...	213	0.001
92	Lordelo do Ouro e Massa...	209	0.001
93	Custóias, Leça do Balio e...	208	0.001
94	Santo Tirso, Couto (Santa...	205	0.001
95	aveleda	202	0.001
96	União das freguesias de ...	201	0.001
97	recarei	200	0.001

Figure A.4: Frequency table 2 of Parish attribute

Index	Nominal value	Absolute count	Fraction
1	porto	187120	0.903
2	Porto	20007	0.097

Figure A.5: Frequency table of District attribute

Index	Nominal value	Absolute count	Fraction
1	porto	47367	0.238
2	vila nova de gaia	35454	0.178
3	gondomar	17292	0.087
4	maia	17137	0.086
5	matosinhos	15868	0.080
6	vila do conde	10412	0.052
7	valongo	9297	0.047
8	póvoa de varzim	8367	0.042
9	Vila Nova de Gaia	3539	0.018
10	paredes	3054	0.015
11	Porto	2867	0.014
12	Gondomar	2670	0.013
13	marco de canaveses	2460	0.012
14	Vila do Conde	2411	0.012
15	Póvoa de Varzim	2322	0.012
16	santo tirso	2182	0.011
17	trofa	2084	0.010
18	penafiel	1867	0.009
19	paços de ferreira	1492	0.008
20	amarante	1400	0.007

Figure A.6: Frequency table of Municipality attribute